.

MPLS Module - Interface Module Description

Chul Kim
Guest Researcher ANTD
E-mail : goldfe@nownuri.net
Alternative: chulkim@antd.nist.gov
Last Update: 2002/12/26
Current State: Draft

Blank

Contents

Figures

1. Interface Module

Since SSFNet simulator provides only NIC interface we have to model Interface module that supports several queuing, scheduling schemes. Interface module manages the interface information such as Interface ID, Interface Type, IP address, and so on. It also contains the resource information and resource management function. In order to enqueue and dequeue a packet, we models the several queue models and scheduler At initialization time, interface management module, MPLSInterfaceMgmt, creates interface module according to the SSFNet interface, such as NIC and ONIC and stores there created interface into hash table.

According to the NIC type we model three different types interface: MPLS NIC interface, MPLS ONIC Interface and MPLS Tunnel Interface. We discuss these interfaces below in detail.

After IP module configures NIC/ONIC interface, management module, MPLSInterfaceMgmt, creates MPLS interface modules with the configured information. Interface management module is one of the main modules of MPLS Module. This module provides the MPLSSignalingInterfaceMgmt module with the resource information and resource management function. It also maintains the Interface table to conserve interface state.

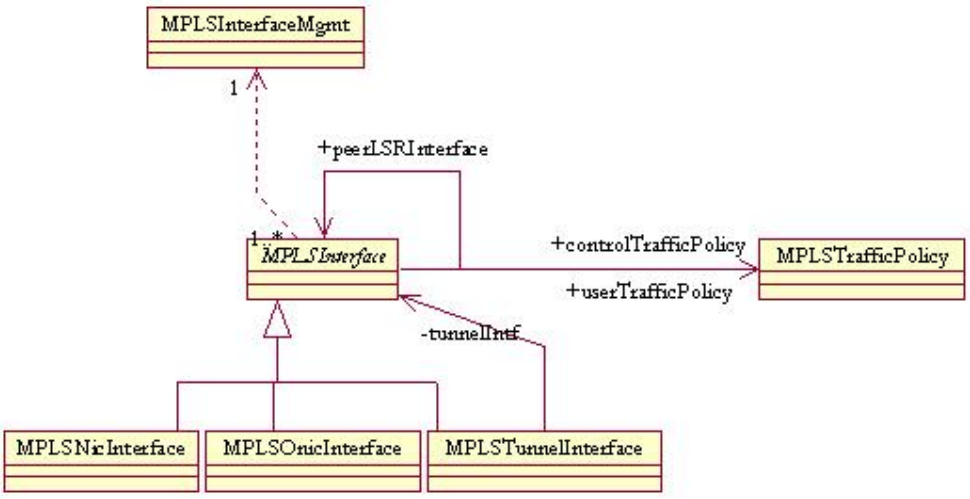

Figure 1. MPLS Interface Class Diagram

Figure 1 shows the class diagram of MPLS Interface modules. The abstract class MPLSInterface is supper class of three different kind interfaces. Each interface has the NIC information and resource management functions inherited from the abstract class. Abstract class, MPLSInterface, has resource management abstract functions and several statistic variables to be used by the monitor module. Figure 2 shows the MPLSInterface abstract class diagram.

Figure 2. MPLSInterface class diagram

MPLSNICInterface module has the information related with NIC module and implements the inherited abstract methods.

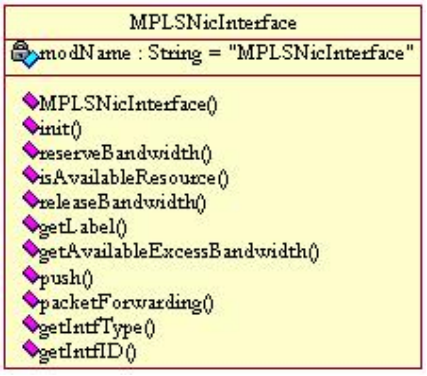

Figure 3. MPLSNicInterface class diagram

MPLS ONIC interface represents the ONIC interface. It has fiber and lambda information to conserve resource information. MPLS Tunnel Interface is created only when the Tunnel LSP is established. Tunnel Interface is only created at Ingress and Egress LSR. Because the tunnel interface is sort of virtual interface, we cannot assign interface ID with NIC ID. We assign the maximum interface ID to the Tunnel Interface ID.

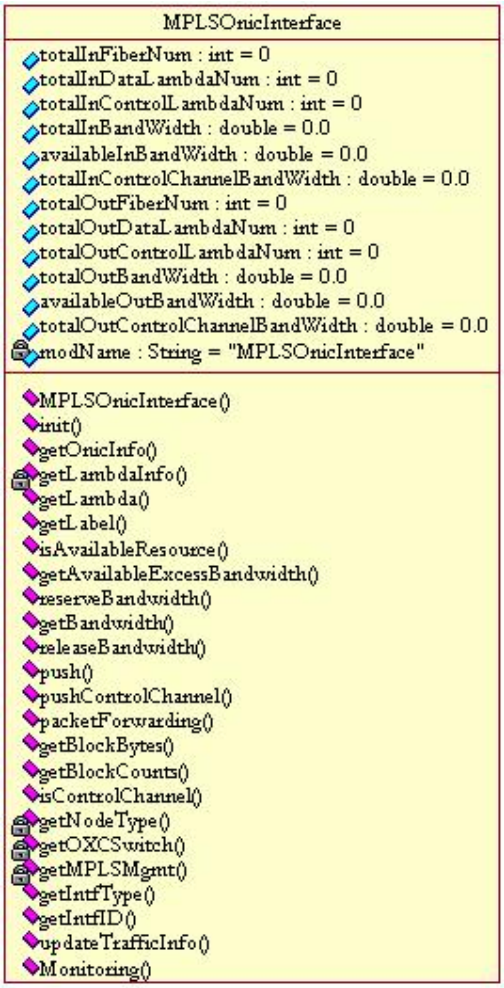

Figure 4. MPLSOnicInterface class diagram

Each MPLSInterface except Tunnel Interface has MPLS Traffic Policy module to shape the incoming traffic It realize the leaky-bucket algorithm with the traffic parameter of LPS. Figure 5 shows the bucket algorithm that is realized in the Policy module. Using the traffic parameter that is delivered by the DML is used to configure three leaky bucket : PBS, CBS, and EBS token bucket.

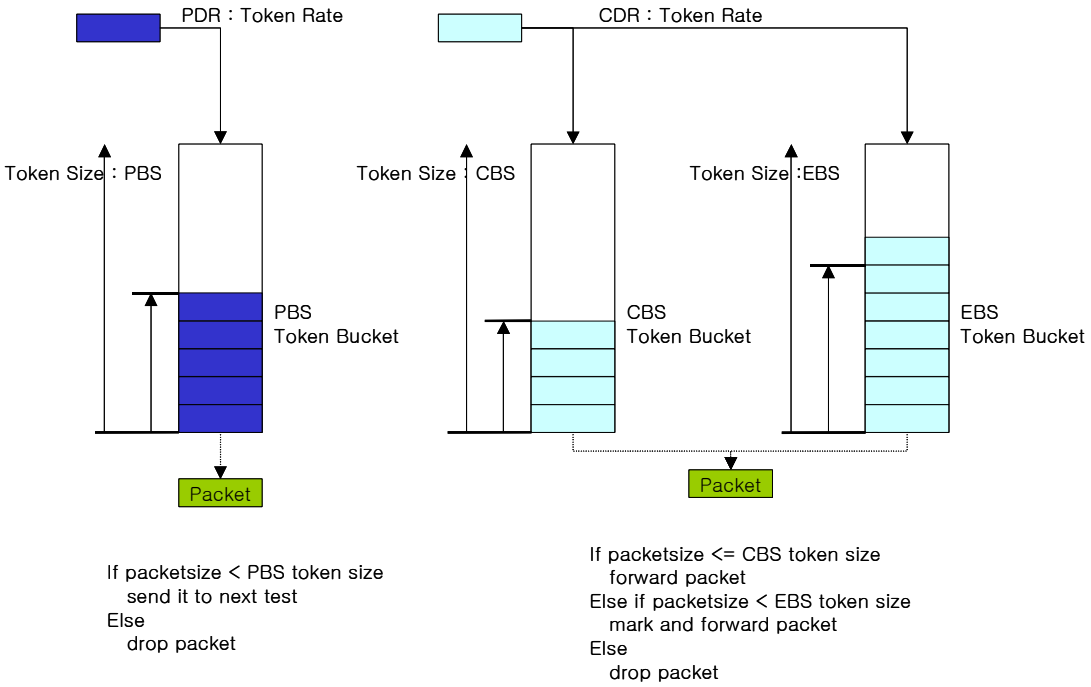

Figure 5 Realized Token Bucket in Traffic Policy module

2. Queue Module

We supports two types of queue: DropTail and RED. We design queue with one class that has ability to be used in Priority scheduler, Weight fair scheduler and round-robin scheduler. The designed queue module, MPLSQueue, is extended from the SimplePackeQueue. This module contains the basic queue information. The designed Queue module has priority field for Priority scheduler, Weight value for Weight Fair Scheduler and queue statistical fields for generating statistical information after simulation.

Figure 6 shows the relationship of MPLSQueue module between the MPLSInterface and MPLSPacketScheduler. Whenever new LSP is established new Queue is created and added to the designated MPLSInterface, which handles it with Hash table. The added queue is also notified to scheduler module to consider this queue's effect on scheduling. MPLSInterface has one-to-many relationship with MPLSQueue as with MPLSPacketScheduler.

The gathered statistical information is maintained in each queue and retrieved by the Queue Monitor module. Queue monitor is specified in DML file and stores the statistical information of each queue in binary file using SSFNet measurement infrastructure. In order to examine binary file we support Queue player that makes directories and files according to the retrieved data from it.

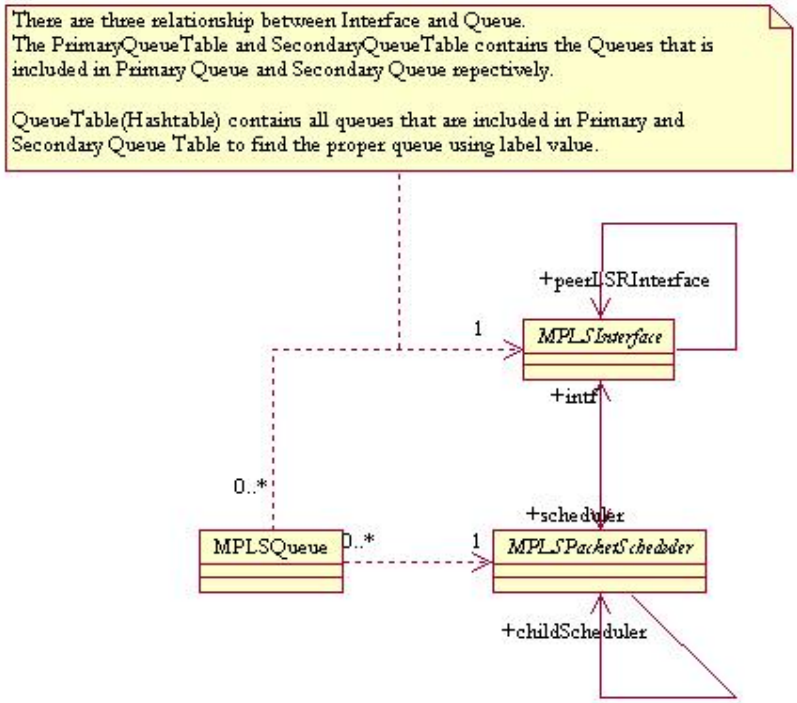

Figure 6. Relationship of MPLSQueue class

3. Scheduler Module

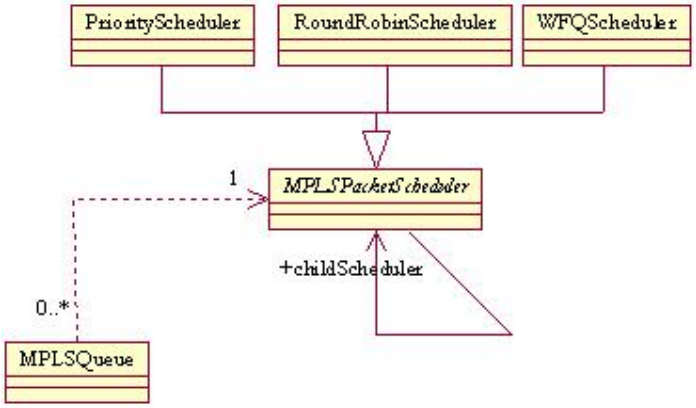

Figure 7. Class Diagram of Scheduler module

Figure 7 shows the class diagram of Scheduler module. We supply three types scheduler: Round-Robin scheduler, Priority Scheduler, and Weight scheduler (based on WFQ). We also support two kinds of scheduler configuration. One is single scheduler configuration the other is hybrid scheduler configuration.

MPLSPackeScheduler is supper-class and three schedulers are derived from it. Priority Scheduler uses priority value of each queue and WFQScheduler uses weight value of queue. To support the hybrid scheduler, MPLSPacketScheduler has own instance as a member variable.